

# Multidimensional slope limiters for MUSCL-type finite volume schemes \*

M.E.Hubbard

The University of Reading, Department of Mathematics,

P.O.Box 220, Whiteknights, Reading, Berkshire, RG6 6AX, U.K.

30th March 1998

## **Abstract**

A number of multidimensional slope limiting operators are presented for MUSCL-type finite volume schemes in two dimensions on triangular grids. The most compressive limiter considered, in the sense that it attains the highest possible magnitude for the reconstructed solution gradient, is implemented in a manner whereby the reconstruction of the solution in a triangle in the limited case is independent of the geometry of the neighbouring cells. Furthermore, the new limiter leads to the definition of multidimensional monotonicity regions within which every other gradient limiter can be placed. The technique is extended to arbitrary polygonal grids, and described in more detail for quadrilaterals. Results are presented for the scalar advection equation, for which an accuracy study is carried out, and for two nonlinear systems, the shallow water and Euler equations. The extension to systems is carried out using Roe's approximate Riemann solver and a comparison is made between the application of the limiter to the conservative variables and to the characteristic variables. All the techniques presented readily generalise to three dimensions.

---

\*This work has been carried out as part of the Oxford/Reading Institute for Computational Fluid Dynamics and was funded by EPSRC.

# 1 Introduction

High order Total Variation Diminishing (TVD) schemes have been developed in one dimension into reliable tools for numerically predicting the solution of hyperbolic systems of equations, including models of convection dominated flows. However, in higher dimensions it has proved difficult to obtain the same degree of robustness and accuracy with extensions of these one-dimensional techniques, particularly on unstructured grids. As a consequence, a great deal of research has been carried out into the generation of genuinely multidimensional high order TVD schemes.

One avenue has led to the development of multidimensional limiting strategies for modifying high order finite volume schemes in such a way that the resulting method is monotonic, thus avoiding the creation of spurious oscillations in the numerical solution and so improving the robustness of the algorithm. Monotonicity is achieved by imposing limits on the solution gradients, obtained via reconstruction of the solution within each cell, so that no new extrema are created by the approximation.

This report addresses a number of issues relating to the application of a specific type of limiter to a simple second order accurate finite volume scheme. The underlying numerical scheme, described in Section 2 for the solution of the scalar advection equation, is a standard upwind cell-centre finite volume method of the MUSCL type with linear reconstruction of the solution within each grid cell [9]. A standard technique for limiting the gradient of the reconstruction to enforce monotonicity [2] is then briefly described, together with a new limiting procedure which is both faster and removes the dependence of the limited reconstruction in a grid cell on the geometry of its neighbouring cells. The description of the new limiter introduces the notion of a monotonicity region for the gradients within which all of the reconstructions must lie to attain monotonicity. The extension of these ideas to quadrilaterals (and general polygonal grids) is also discussed, and the resulting schemes compared with a standard operator split scheme. Results are presented for each of the schemes described.

Section 3 describes the extension of these methods to solve nonlinear systems

of equations, specifically the shallow water equations and the Euler equations. The basic high order scheme is described, in which Roe's approximate Riemann solver is applied at the grid edges to decompose the system into components for which the scalar scheme can be used. The limiting procedure is carried out on both the conservative variables and the characteristic variables in turn and the results compared.

## 2 The Scalar Advection Equation

In conservation form the two-dimensional scalar advection equation is written

$$u_t + f_x + g_y = 0, \quad (2.1)$$

where the conservative fluxes  $f = f(u)$  and  $g = g(u)$  are functions of the solution variable  $u$ .

A MUSCL type cell-centre finite volume method for the numerical solution of the scalar advection equation is described as follows. Integrate (2.1) over a control volume,  $\Omega$  say, (here taken to be a single grid cell) and apply the divergence theorem to the resulting flux integral, giving the equation

$$\int \int_{\Omega} u_t \, dx \, dy + \oint_{\partial\Omega} \vec{f} \cdot d\vec{n} = 0, \quad (2.2)$$

where  $\vec{f} = (f, g)^T$  is the flux function and  $\vec{n}$  represents an outward pointing normal to the boundary  $\partial\Omega$  of the control volume.

Approximation of the boundary integral in (2.2) leads to the finite volume discretisation

$$\frac{\partial \bar{u}}{\partial t} = -\frac{1}{V} \sum_{k=1}^{N_e} \vec{f}_k^* \cdot \vec{n}_k, \quad (2.3)$$

in which  $\bar{u}$  is defined to be the average value of  $u$  over the control volume  $\Omega$ ,  $V$  is the area of the control volume,  $N_e$  is the number of its edges and  $\vec{n}_k$  is the outward pointing normal to the  $k^{\text{th}}$  edge scaled by its length. Note that since the control volumes coincide with the grid cells, the numerical flux function  $\vec{f}_k^*$  is an approximation to the flux at a particular grid edge.

Assuming that the approximation to  $u$  is piecewise constant within each cell and generally discontinuous at the cell edges, as illustrated in Figure 2.1, a scheme

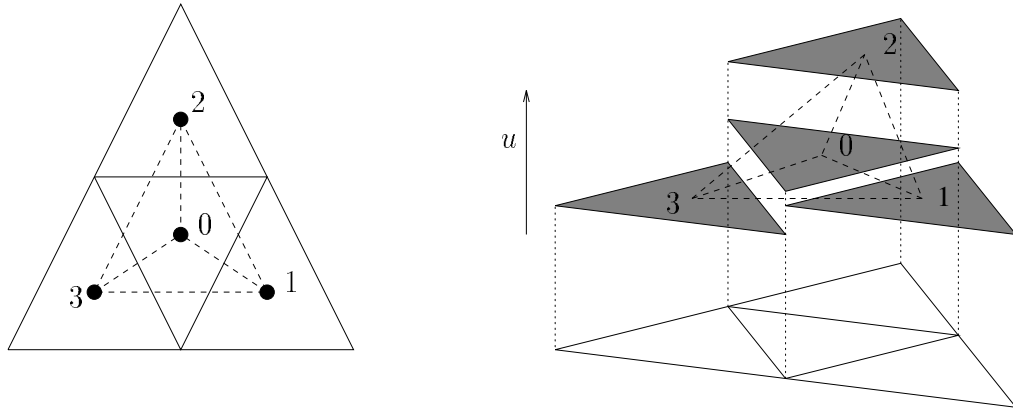


Figure 2.1: The limiting planes as defined for a triangular control volume (left) and a piecewise constant reconstruction of the solution (right).

which is first order accurate in space is obtained. The scheme can be made monotonic by introducing an upwind bias into the evaluation of the numerical flux function. The word monotonic is slightly ambiguous when used in a two-dimensional context but will be used from now on to denote a scheme which doesn't create spurious extrema at the new time level. Taking as an example the  $k^{\text{th}}$  edge of cell 0 in Figure 2.1, the upwinding is applied [5] by defining

$$\vec{f}^*(u_0, u_k) \cdot \vec{n}_k = \begin{cases} u_0 \tilde{\lambda} \cdot \vec{n}_k & \text{if } \tilde{\lambda} \cdot \vec{n}_k \geq 0 \\ u_k \tilde{\lambda} \cdot \vec{n}_k & \text{otherwise,} \end{cases} \quad (2.4)$$

where  $u_k$  is the value of  $u$  in the adjacent grid cell and  $\tilde{\lambda}$  is an appropriate average of the advection velocity vector  $\vec{\lambda} = \left( \frac{\partial f}{\partial u}, \frac{\partial g}{\partial u} \right)$  evaluated from the solution values  $u_0$  and  $u_k$ . In the special case of constant advection,

$$\tilde{\lambda} = \vec{\lambda} = \begin{pmatrix} a \\ b \end{pmatrix} \quad (2.5)$$

where  $a$  and  $b$  are constant throughout the domain. Note that an equivalent expression to (2.4) is given by

$$\vec{f}^*(u_0, u_k) \cdot \vec{n}_k = \frac{1}{2}(\vec{f}_0 + \vec{f}_k) \cdot \vec{n}_k - \frac{1}{2}|\tilde{\lambda} \cdot \vec{n}_k|(u_k - u_0) \quad (2.6)$$

which is preferred in the generalisation to nonlinear systems of equations because of its symmetry. Furthermore, although only triangular grid cells are illustrated in Figure 2.1, Equations (2.4) and (2.6) are valid for any polygonal grid cells, such as quadrilaterals.

## 2.1 Gradient Operators and Higher Order Schemes

Higher order spatial accuracy is achieved here by introducing a higher order reconstruction of the variable  $u$  within each grid cell. For example, a piecewise linear approximation to the solution, such as that shown in Figure 2.2, which is exact for linear initial data, leads to a second order method.

Thus, given an initial constant (or average) solution value  $\bar{u}$  within a cell we carry out a linear reconstruction of  $u$  within that cell expressed as

$$u = \bar{u} + \vec{r} \cdot \vec{L} , \quad (2.7)$$

where  $\vec{r}$  is a position vector relative to the centroid of the cell and  $\vec{L}$  is a gradient operator, yet to be defined. It is easy to show that such a reconstruction is conservative in the sense that

$$\frac{1}{V} \int \int_{\Omega} u \, dx \, dy = \bar{u} . \quad (2.8)$$

It can also be shown [2] that when (2.8) is satisfied the resulting numerical scheme (2.3) can be guaranteed to be monotonic for an appropriate restriction on the time-step as long as the reconstruction (2.7) within each cell does not lead to the creation of any new extrema at the *midpoints* of the edges of that cell.

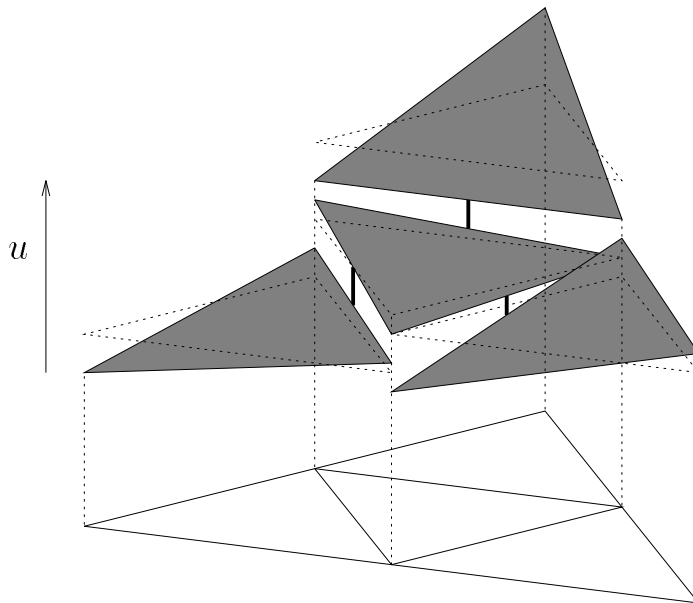


Figure 2.2: A piecewise linear reconstruction of the solution for a triangular control volume.

In the present case the numerical flux function of Equation (2.6) at a cell edge, such as one of those shown in Figure 2.2, is written in terms of the reconstructed solution values from the two neighbouring cells at the midpoint of the edge, so that  $\vec{f}^*$  is now written

$$\vec{f}^*(u_L, u_R) = \vec{f}^*(u_0 + \vec{r}_{0k} \cdot \vec{L}_0, u_k + \vec{r}_{k0} \cdot \vec{L}_k), \quad (2.9)$$

where  $\vec{r}_{ij}$  is the vector from the centroid of cell  $i$  to the midpoint of the edge between cells  $i$  and  $j$ , and  $\vec{L}_i$  is the gradient of the reconstructed solution in cell  $i$ . In the notation used here  $u_L$  is considered to be an *interior* reconstructed solution value relative to the cell under consideration and  $u_R$  is the corresponding *exterior* value, taken from the adjacent cell which is generally different. This motivates the subsequent use of a Riemann solver to evaluate the edge fluxes. It now remains to define an appropriate gradient  $\vec{L}$  for the linear reconstruction of the solution within each cell of the grid.

A simple gradient operator which is exact for linear data can be defined on any grid by taking the (constant) solution value in three arbitrarily chosen, but preferably adjacent, cells ( $i, j$  and  $k$  say, forming a triangle with anticlockwise indexing of its vertices) and defining

$$\vec{\nabla}(\triangle ijk) = \begin{cases} \begin{pmatrix} -\frac{n_x}{n_u} \\ -\frac{n_y}{n_u} \end{pmatrix} & \text{for } n_u \geq \epsilon \\ \begin{pmatrix} 0 \\ 0 \end{pmatrix} & \text{otherwise,} \end{cases} \quad (2.10)$$

in which  $\epsilon \approx 10^{-10}$  and  $n_x, n_y$  and  $n_u$  are the components of the vector  $\mathbf{n}$  normal to the plane defined by the triangle  $ijk$  in  $xyu$ -space given by

$$\mathbf{n} = (\underline{P}_i - \underline{P}_k) \times (\underline{P}_j - \underline{P}_k), \quad (2.11)$$

where

$$\underline{P}_* = \begin{pmatrix} x_* \\ y_* \\ u_* \end{pmatrix}. \quad (2.12)$$

The vector  $\vec{n}$  has been constructed in such a way that  $n_u$  always has the same sign as the area of  $\triangle ijk$ , so the second option in (2.10) has been introduced to

deal with the possibility of  $\Delta ijk$  having a non-positive area. An example of such a triangle is shaded in Figure 2.3.

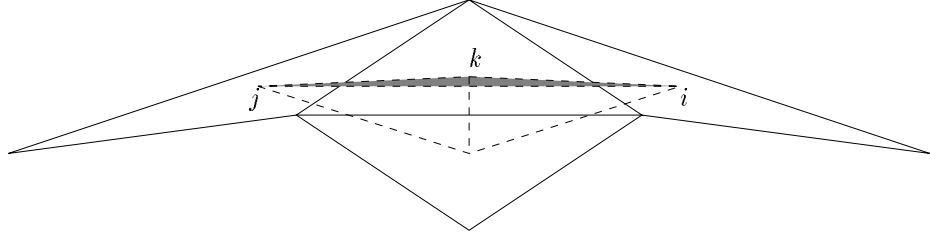


Figure 2.3: A reconstruction triangle with negative area (shaded).

Selecting  $\vec{L}$  in (2.9) to be the  $\vec{\nabla}$  operator of (2.10) leads to a second order accurate method but doesn't prohibit overshoots and undershoots at the midpoints of the cell edges and therefore the scheme does not satisfy the monotonicity condition. In order to impose monotonicity the gradient operator  $\vec{L}$  must be defined to be a 'limited' form of  $\vec{\nabla}$ .

### 2.1.1 Gradient Limiters

Previously constructed gradient limiters [2, 3] applied to the above scheme have required

- a) the construction of a gradient operator of the form (2.10) from which to choose the (unlimited) reconstruction of the solution.
- b) the limiting of the chosen gradient operator to impose monotonicity.

On a triangular grid the simplest approach, known as the Limited Central Difference (LCD) scheme, considers only  $\vec{\nabla}(\Delta 123)$  (in the notation of Figure 2.1) in step a) above. The limiting stage involves the calculation of

$$\alpha^k \leq \begin{cases} \frac{\max(u_k - u_0, 0)}{\vec{r}_{0k} \cdot \vec{L}} & \text{if } \vec{r}_{0k} \cdot \vec{L} > \max(u_k - u_0, 0) \\ \frac{\min(u_k - u_0, 0)}{\vec{r}_{0k} \cdot \vec{L}} & \text{if } \vec{r}_{0k} \cdot \vec{L} < \min(u_k - u_0, 0) \\ 1 & \text{otherwise} \end{cases} \quad (2.13)$$

for each edge  $k$ . The LCD limited gradient is then given by

$$\vec{L} = \alpha \vec{\nabla}(\Delta 123) = \left( \min_{k=1,2,3} \alpha^k \right) \vec{\nabla}(\Delta 123). \quad (2.14)$$

The Maximum Limited Gradient (MLG) scheme devised by Batten *et al.* [2] improves on the LCD scheme by using the ideas of Durlofsky *et al.* [3] to create a far more compressive limiter. The MLG scheme chooses its initial reconstruction from the four gradient operators

$$\vec{\nabla}(\Delta 123), \quad \vec{\nabla}(\Delta 023), \quad \vec{\nabla}(\Delta 103), \quad \vec{\nabla}(\Delta 120). \quad (2.15)$$

Each of these is limited using the procedure described by (2.13) and (2.14), leading to the limited gradient planes

$$\begin{aligned} \vec{L}_0 &= \alpha_0 \vec{\nabla}(\Delta 123), & \vec{L}_1 &= \alpha_1 \vec{\nabla}(\Delta 023), \\ \vec{L}_2 &= \alpha_2 \vec{\nabla}(\Delta 103), & \vec{L}_3 &= \alpha_3 \vec{\nabla}(\Delta 120). \end{aligned} \quad (2.16)$$

The MLG limited gradient operator is then taken to be the  $\vec{L}_i$  of (2.16) with the largest slope  $|\vec{L}_i|$ . The main disadvantage of this limiter is the extra expense involved in computing and limiting four gradient planes rather than just one, although it is significantly more accurate than the LCD scheme.

In one dimension the MLG limiter reduces to the well-known Superbee limiter [8]. Note though that this two-dimensional generalisation does not depend continuously on the data. It is simple to construct a situation where the value of  $u_0$  is allowed to vary continuously (with  $u_1$ ,  $u_2$  and  $u_3$  remaining fixed) but the limited gradient operator in (2.16) varies discontinuously as the operator on which it is based switches between the planes defined in (2.15). This is most significant close to a steady state solution when the discontinuity may interfere with convergence by causing limit cycling. The limiter of Durlofsky *et al.* [3] suffers from the same problem.

It is worth commenting that steps a) and b) of the limiting procedure, as described at the start of this section, can easily be extended to arbitrary polygonal/polyhedral control volumes in two and three dimensions. This is useful if general polygonal or hybrid grids are to be used, or even for highly distorted quadrilateral grids on which the standard directionally split techniques give poor accuracy.

On quadrilaterals, for example, an MLG-type limiter may be constructed in essentially the same manner as for triangles, the main difference being in the



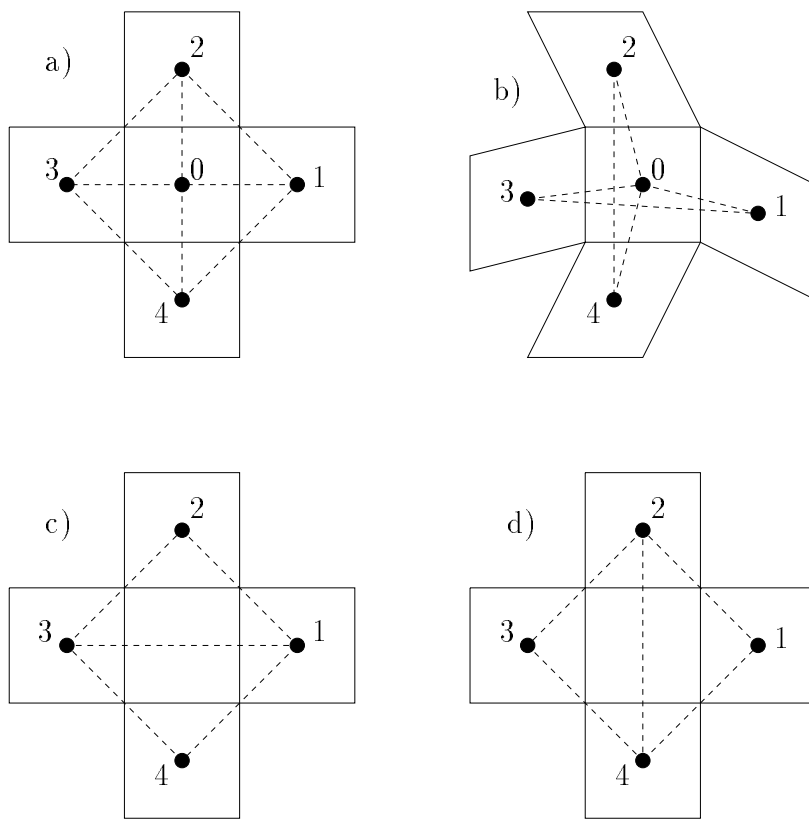


Figure 2.4: The limiting planes as defined for a quadrilateral control volume.

selection of the gradient operators, *cf.* (2.14) and (2.15) in the first step of the limiting. A grid cell now has four edge-neighbours and hence ten possible choices for the gradient operator (all shown in Figure 2.4). For the sake of computational speed not all of these operators would be considered. In particular, the two triangles in Figure 2.4b) have no natural order to their vertices; this introduces an ambiguity into the definition of negative areas and so it is natural to exclude them from the algorithm. In general there are  $C_3^n$  possible choices of gradient plane for a shape with  $n$  edges/faces. The limiting procedure (2.16) can then be applied as before to the chosen operators. This technique readily carries over to general polygonal and polyhedral grids in two and three dimensions, even if adaptive grid refinement creates hanging nodes.

### 2.1.2 A ‘Maximum Slope’ Limiter

Consider a triangular cell, as illustrated in Figure 2.1, and its three edge-neighbours. Each of the gradient operators of (2.15) defines a direction

$$\widehat{\vec{L}}_i = \frac{\vec{\nabla}(\Delta_{ijk})}{|\vec{\nabla}(\Delta_{ijk})|}, \quad (2.17)$$

in which the reconstructed gradient is to be maximised subject to the creation of no new extrema at the midpoints of the cell edges. The gradient operator which is ultimately chosen by the MLG scheme is the plane from (2.15) which allows the ‘steepest’ limited slope. However, there is only a finite number of gradient planes to choose from, there is no reason why the algorithm should have picked out the steepest monotonic reconstruction possible. Only this reconstruction will lead to the most compressive limiter.

The monotonic linear reconstruction  $\vec{L}$  with maximal gradient can be calculated in a cell 0 from the following simple constrained optimisation problem:

- Maximise  $|\vec{L}|$  subject to

$$\min(u_k - u_0, 0) \leq \vec{r}_{0k} \cdot \vec{L} \leq \max(u_k - u_0, 0) \quad (2.18)$$

for  $k = 1, 2, 3$ , where  $\vec{r}_{0k}$  is the vector from the centroid of cell 0 to the midpoint of the edge between cells 0 and  $k$ .

For clarity, this may be rewritten as

- Maximise  $f(x, y) = x^2 + y^2$  subject to

$$\min(u_k - u_0, 0) \leq (r_x)_{0k} x + (r_y)_{0k} y \leq \max(u_k - u_0, 0) \quad (2.19)$$

for  $k = 1, 2, 3$ . The desired gradient operator is then given by  $\vec{L} = (x, y)^T$ .

### 3

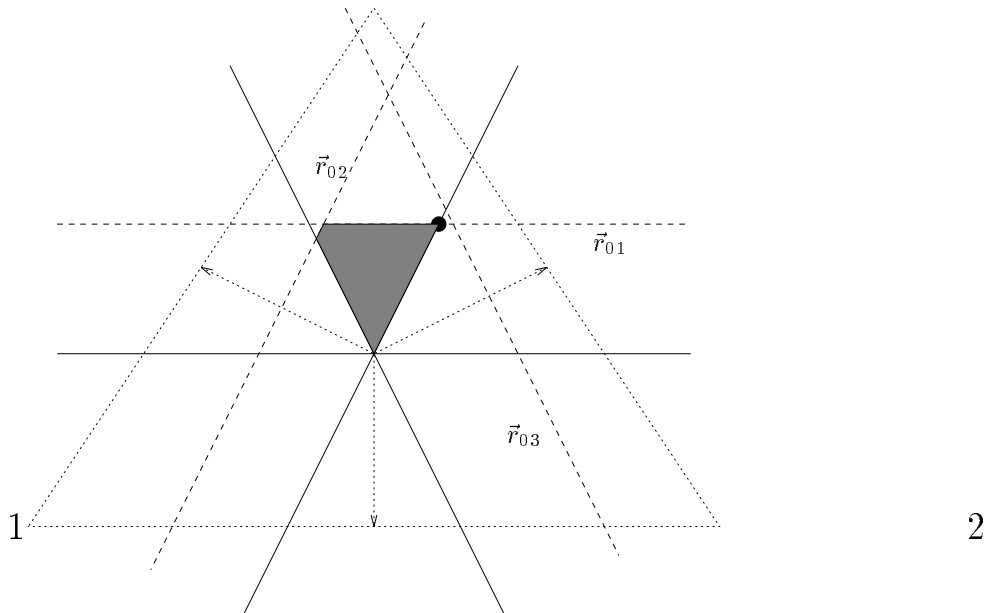


Figure 2.5: An example of a monotonicity region (shaded) - the black dot indicates the position representing the steepest monotonic gradient plane.

The solution of this optimisation problem is illustrated in Figure 2.5. The shaded area indicates the region within which  $f(x, y)$  satisfies the criteria (2.19), from now on called the monotonicity region. Since the family of curves given by  $f(x, y) = \text{constant}$  defines a set of concentric circles then  $f(x, y)$  is maximised at an intersection of the boundary edges of the monotonicity region. In the case of the MLG limiter, four further constraints are added;  $f(x, y)$  is still maximised but now it must not only satisfy (2.19) but also the resulting gradient operator  $\vec{L}$  must be parallel to one of (2.15). There are infinitely many ways in which the limited gradient operator but no others will be discussed here.

For an arbitrary triangular grid and solution data there are at most ten distinct intersections of the straight lines defined by the constraint inequalities (2.19), including  $\vec{L} = \vec{0}$  (when  $x = y = 0$ ) which is optimal only when  $u_0$  is a local extremum. Generally there is a maximum of  $\frac{3}{2}n(n-1) + 1$  possible solutions for an  $n$ -sided polygon. Although more sophisticated techniques can be used,  $\vec{L}$  is optimised here by calculating each feasible solution and choosing the appropriate one. With a more complicated logical structure only one gradient operator need be evaluated in each cell.

Although the optimisation problem proves to be relatively expensive to solve, it need only be considered in cells where the initial reconstruction gives rise to overshoots and undershoots. Furthermore, the limiter does not have to be applied explicitly if the cell value  $u_0$  is detected to be a local extremum *a priori* since then  $\vec{L} = \vec{0}$ . Hence the additional expense is only incurred in a small proportion of the grid cells. In fact the implementation described below is significantly cheaper in practice than the MLG algorithm.

It should also be noted that the new limiting procedure depends only on the values of the solution in the edge-neighbours of a cell and not on the position of their centroids, removing one aspect of grid dependence from the algorithm.

The Grid-Independent (GI) limiter on a triangular grid can be summarised as follows:

- Calculate the gradient operator  $\vec{\nabla}(\Delta 123)$  as in the LCD scheme and check whether it creates any new local extrema at the edge midpoints.
  - if it doesn't, set  $\vec{L} = \vec{\nabla}(\Delta 123)$ .
  - otherwise calculate the monotonic gradient operator  $\vec{L}$  with maximal gradient from (2.19).

In fact all four gradient operators of the MLG limiter (2.15) could be checked for monotonicity, but the small increase in compression is not worth the extra computational expense (even though it would then reduce to the Superbee limiter in one dimension).

As it stands, the gradient operators calculated by the above algorithm do not depend continuously on the data. This can be corrected in the case when

only a single gradient operator is considered by constructing a transition function which imposes a continuous change between the unlimited gradient operator  $\vec{L}_0 = \vec{\nabla}(\Delta 123)$  and the optimal gradient operator  $\vec{L}_{\text{opt}}$  calculated from (2.19). This resulting gradient operator is taken to be

$$\vec{L} = \alpha_0 \vec{L}_0 + \frac{1 - \max(\alpha_0, 1 - \epsilon)}{\epsilon} (\vec{L}_{\text{opt}} - \alpha_0 \vec{L}_0), \quad (2.20)$$

where  $0 \leq \epsilon < 1$  (typically  $\epsilon = 0.01$ ) and  $\alpha_0$  is the limiting factor associated with  $\vec{L}_0$  as defined in (2.13). The important property of this operator is that

$$\vec{L} = \begin{cases} \vec{L}_0 & \text{when } \alpha_0 = 1 \\ \vec{L}_{\text{opt}} & \text{when } \alpha_0 = 1 - \epsilon. \end{cases} \quad (2.21)$$

In practice, this additional restriction appears to be unnecessary for time-dependent applications.

As with the MLG limiter, the GI limiter in its discontinuous form is easily generalised to other polygonal shapes of control volume. For example, it can be seen that on a two-dimensional quadrilateral grid where the control volumes coincide with the grid cells the differences from the algorithm on triangles are in the initial choice of gradient operators (*cf.* Figure 2.4) and the inclusion of a fourth monotonicity constraint in (2.18) and (2.19). Unfortunately, the search for the optimal direction is complicated considerably by the addition of this extra equation. There are now 19 feasible solutions for the resulting optimisation problem, each of which must be checked for its optimality. This is not straightforward, particularly on regular grids since the region satisfying the inequalities often reduces to a straight line which can be difficult to detect. This problem could not arise on a triangular grid.

## 2.2 A Directionally Split Limiter

On structured quadrilateral grids it is easy to construct a directionally split limiter, *e.g.* in two dimensions a one-dimensional limiter is applied in two independent directions determined by the grid. The Directionally Split (DS) limiting procedure described in [1] is used here to construct an operator splitting method to compare with the multidimensional limiters described above.

The DS limiter is similar in many ways to the slope limiters described above, and the multidimensional gradient operators can be modified and limited so that the resulting scheme is equivalent on rectangular grids. The gradient operators are decoupled into components in the two grid directions and a separate value of  $\alpha$  is calculated in the manner of (2.13) (taking into account only the constraints in the relevant grid directions) for each of the components, so

$$\vec{L} = \begin{pmatrix} \alpha_x L_x \\ \alpha_y L_y \end{pmatrix}. \quad (2.22)$$

The MLG-type limiting procedure calculates a single value of  $\alpha$  which satisfies

$$\alpha = \min(\alpha_x, \alpha_y) \quad (2.23)$$

and is therefore far less compressive.

The type of modification given by (2.22) cannot be applied on grids where successive cell centroids do not lie in a straight line, but then the DS approach as described in [1] is not actually linearity preserving on such grids since the grid ‘directions’ become meaningless.

The GI approach is more flexible since it only requires a short calculation to check whether new extrema have been created by the chosen gradient operators. If so then the optimal gradient operator is constructed from (2.19) which, on rectangular grids, is equivalent to the gradient operator derived from the split approach above (2.22). Although the GI scheme is linearity preserving on any mesh, the extra speed and in general higher compression of the DS scheme gives it a considerable advantage on structured grids.

### 2.3 Boundaries

The limiting procedure is applied very simply at boundaries of the domain. In step a) of the limiting procedure only those gradient operators which can be constructed from centroids of control volumes *within* the domain are included and the others are assumed to be zero. Also, only internal solution values are considered in the search for new extrema in the reconstruction. On a triangular grid this means that only a single gradient operator is constructed (and limited)

in each cell with just one boundary edge. The scheme therefore produces an exact reconstruction of linear data on triangles except in cells with multiple boundary edges. When periodic boundary conditions are used no special treatment of the boundaries is needed.

On structured quadrilateral grids at least one gradient operator can always be constructed in each cell, so the scheme reconstructs linear data exactly throughout the domain.

## 2.4 Time Integration

Second order temporal accuracy may be obtained using a Runge-Kutta time-stepping method such as that given by

$$\begin{aligned}
\bar{u}_0 &= u_0^n - \frac{\Delta t}{V} \sum_{k=1}^{N_e} \vec{f} \left( u_0^n + \vec{r}_{0k} \cdot \vec{L}_0^n, u_k^n + \vec{r}_{k0} \cdot \vec{L}_k^n \right) \cdot \vec{n}_k \\
u_0^{n+1} &= \frac{1}{2} \left( u_0^n + \bar{u}_0 - \frac{\Delta t}{V} \sum_{k=1}^{N_e} \vec{f} \left( \bar{u}_0 + \vec{r}_{0k} \cdot \vec{L}_0, \bar{u}_k + \vec{r}_{k0} \cdot \vec{L}_k \right) \cdot \vec{n}_k \right) \\
&= u_0^n - \frac{\Delta t}{2V} (\delta u_0 + \delta \bar{u}_0) .
\end{aligned} \tag{2.24}$$

However, the cost of the reconstructions and the local Riemann solutions is prohibitively expensive, so the following approximation to the above update scheme [10] is used instead:

$$\begin{aligned}
\bar{u}_0 &= u_0^n - \frac{\Delta t}{2V} \sum_{k=1}^{N_e} \vec{f} \left( u_0^n + \vec{r}_{0k} \cdot \vec{L}_0^n \right) \cdot \vec{n}_k \\
u_0^{n+1} &= u_0^n - \frac{\Delta t}{V} \sum_{k=1}^{N_e} \vec{f} \left( \bar{u}_0 + \vec{r}_{0k} \cdot \vec{L}_0^n, \bar{u}_k + \vec{r}_{k0} \cdot \vec{L}_k^n \right) \cdot \vec{n}_k .
\end{aligned} \tag{2.25}$$

It has been shown [2] that on triangular grids any limiter of the type described in this report is monotonic for a restriction on the time-step within each cell given by

$$\Delta t \leq \frac{V}{3 \max_k |\vec{\lambda} \cdot \vec{n}_k|} . \tag{2.26}$$

It can also be shown easily that the corresponding restriction necessary for monotonicity to be satisfied on quadrilateral grids is

$$\Delta t \leq \frac{V}{4 \max_k |\vec{\lambda} \cdot \vec{n}_k|} . \tag{2.27}$$

In both of the above limits the maximum is taken over the adjacent cells.

The global time-step restriction used with the DS limited scheme is defined to be

$$\Delta t \leq \min_{i,j} \frac{|\vec{x}_{ij}|}{2 (c + \sqrt{u^2 + v^2})_{ij}}, \quad (2.28)$$

where  $\vec{x}_{ij}$  is the vector joining the centroid of cell  $i$  to the centroid of cell  $j$ , an edge-neighbour of cell  $i$ .

## 2.5 Results

Numerical experiments have been carried out to test the behaviour of all of the schemes described in this report. The first test presented here is the advection of an initial profile given by the double sine wave function

$$u = \sin(2\pi x) \sin(2\pi y), \quad (2.29)$$

with velocity  $\vec{\lambda} = (1, 1)^T$  over the domain  $[0, 1] \times [0, 1]$ . This problem has been solved on three types of grid each of which is illustrated in Figure 2.6. Periodic boundary conditions are applied.

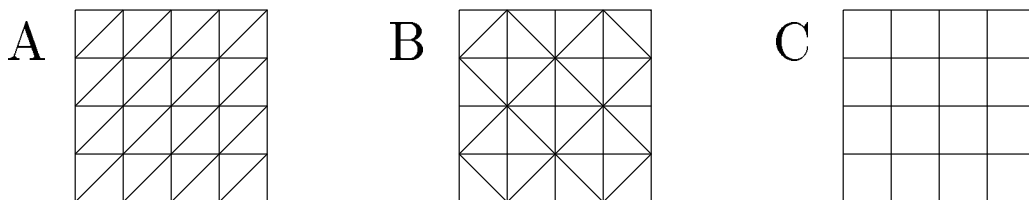


Figure 2.6: The three grid types used for the numerical experiments.

Error estimates in the  $L_2$  norm for the solution when  $t = 0.1$  are shown in Figure 2.7. The errors in the  $L_1$  and  $L_\infty$  norms behave in a similar manner for all of the grids used. In all of the numerical experiments the ratio  $dt/dx = 0.04$  where  $dx$  is the length of any horizontal grid edge in Figure 2.6. The first order scheme is unsurprisingly the least accurate. Of the second order schemes the MLG limiter gives easily the most accurate results on grid A and is the only one of the schemes on triangles that approaches second order accuracy (roughly 1.88 at the finest grid level), better even than the dimensionally split quadrilateral scheme for which errors are also shown in the figure. It should be noted that for



a precise comparison in terms of computational effort (*i.e.* number of cells rather than grid size) the graph for the quadrilateral grid should be shifted 0.15 to the right. Even then the MLG scheme is better. On the coarsest grid the GI scheme is best but it rapidly deteriorates until it is only slightly more accurate than the LCD scheme.

On grid B all of the linear reconstruction schemes on triangles give considerably worse results than on grid A, in particular the MLG scheme, which is now the worst on the finest grids. It is now the GI scheme which is the most accurate, although the results are nowhere near the quality of those obtained on the quadrilateral grids.

The advantage of using the GI scheme is clarified by considering a second test case. It involves the circular advection of the ‘cone’, given by the initial conditions (when  $t = 0$ )

$$u = \begin{cases} \cos^2(2\pi r) & \text{for } r \leq 0.25 \\ 0 & \text{otherwise} \end{cases} \quad (2.30)$$

where  $r^2 = (x + 0.5)^2 + y^2$ , with velocity  $\vec{\lambda} = (-2\pi y, 2\pi x)^T$  around the domain  $[-1, 1] \times [-1, 1]$ , the solution being continually set to zero at each of the inflow boundaries. The initial profile should be advected in a circle without change of shape until it returns to its original position when  $t = 1.0$ .

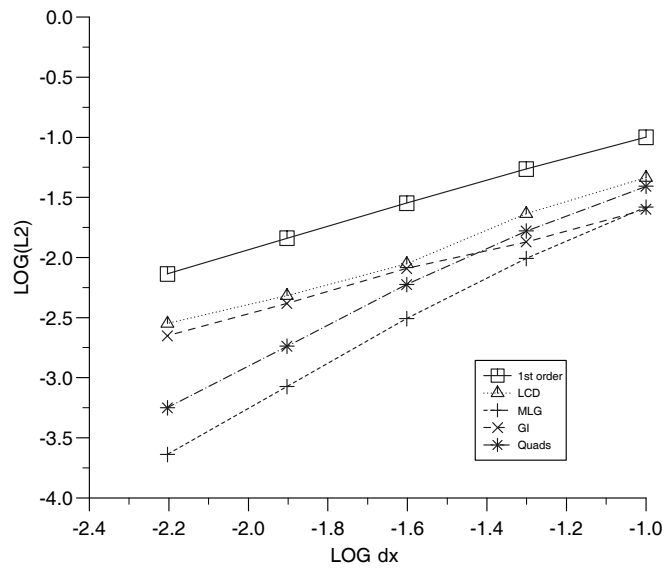
Four solution profiles obtained on a  $64 \times 64$  type B grid are shown in Figures 2.8 and 2.9. Of the four, the GI scheme clearly retains the peak best, followed by the MLG, LCD and first order schemes in decreasing order of accuracy. The errors in the solutions are shown in Table 1, along with computing times, and these clearly show the GI scheme to be the most accurate for this test case. The table also shows that the GI scheme, although slower than the LCD scheme, is about 12% faster than the MLG method.

### 3 Systems of Equations

The extension of these cell centred finite volume schemes to nonlinear systems of equations is straightforward. The conservative equations take the general form

$$\underline{U}_t + \underline{F}_x + \underline{G}_y = \underline{Q}, \quad (3.1)$$

## Grid type A



## Grid type B

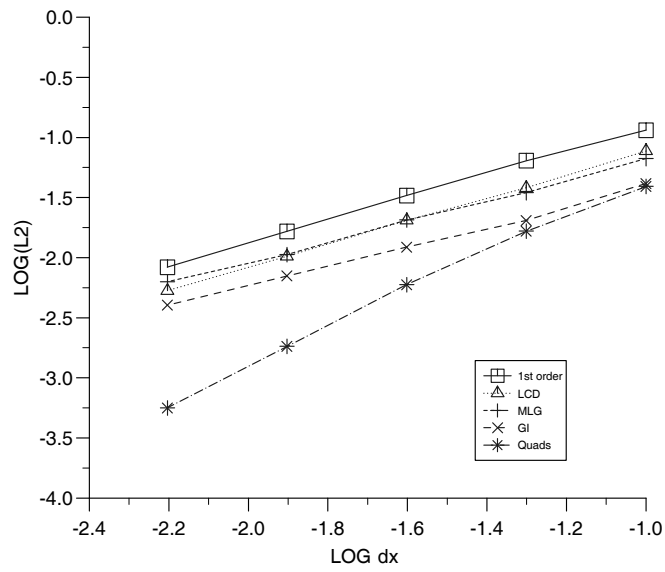
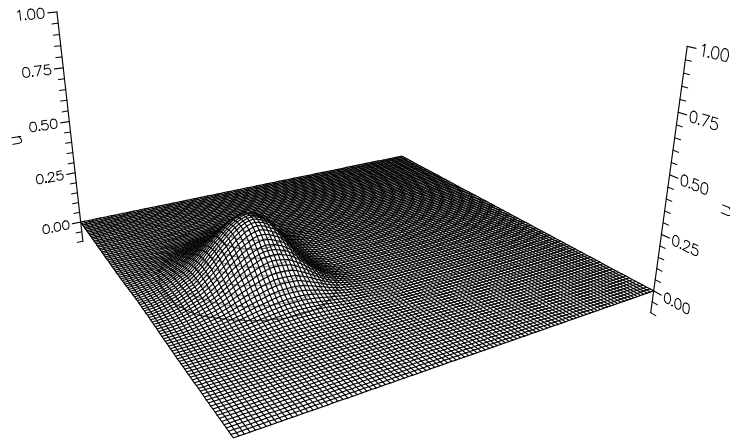


Figure 2.7:  $L_2$  errors for the double sine wave test case.

## First order scheme



## LCD scheme

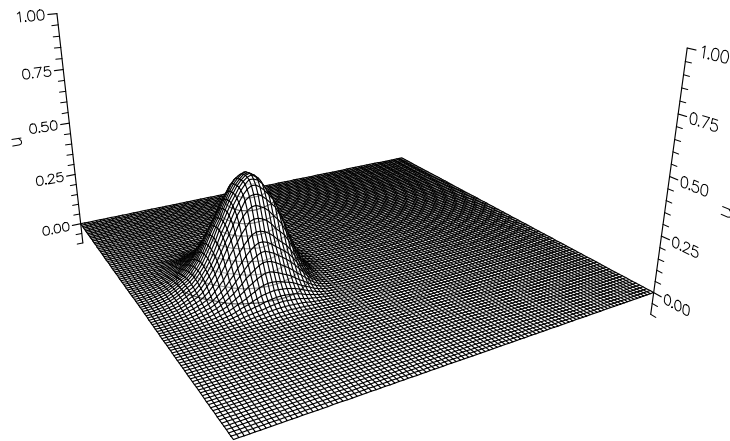
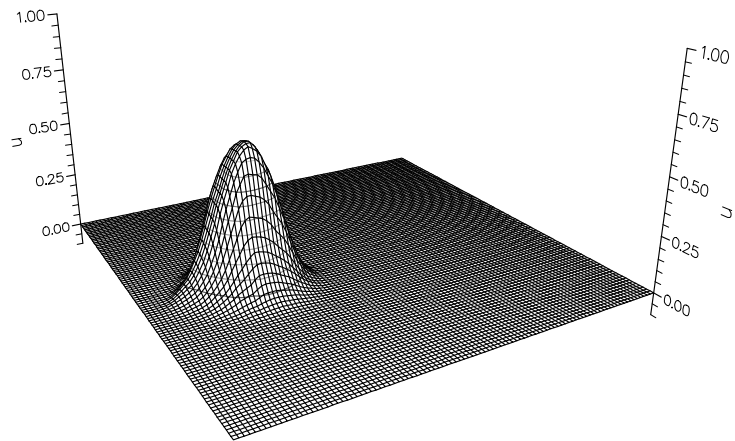


Figure 2.8: Solutions for the rotating cone test case.

## MLG scheme



## GI scheme

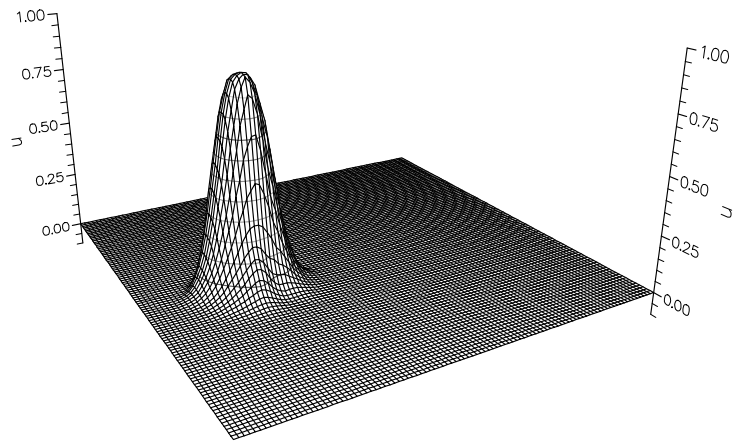


Figure 2.9: Solutions for the rotating cone test case.

Scheme	$L_1$	$L_2$	$L_\infty$	Peak value	Time (s)
First order	0.0152	0.0621	0.7161	0.284	94
LCD	0.0094	0.0431	0.5358	0.479	120
MLG	0.0055	0.0281	0.3855	0.617	154
GI	0.0029	0.0152	0.2009	0.887	137

Table 1: Error norms of solutions to the rotating cone problem at  $t = 1.0$  on a  $64 \times 64$  type B grid.

in which  $\underline{U}$  is the vector of conserved variables and  $\underline{F}$ ,  $\underline{G}$  are the conservative flux vectors. These are defined explicitly for the shallow water equations and the Euler equations in Appendices A and B respectively.

Integrating the equations (3.1) over a control volume  $\Omega$  (taken as before to be a grid cell) and applying the divergence theorem to the flux integral results in

$$\iint_{\Omega} \underline{U}_t \, dx \, dy + \oint_{\partial\Omega} (\underline{F}, \underline{G}) \cdot d\vec{n} = 0, \quad (3.2)$$

where  $\vec{n}$  again represents an outward pointing normal to the boundary. Approximating the boundary integral and defining  $\underline{U}_0$  to be the average value of  $\underline{U}$  over the control volume  $\Omega$  leads to the finite volume discretisation

$$\frac{\partial \underline{U}_0}{\partial t} = -\frac{1}{V} \sum_{k=1}^{N_e} (\underline{F}^*, \underline{G}^*) \cdot \vec{n}_k, \quad (3.3)$$

where  $V$  is the area of the control volume,  $N_e$  is the number of edges of the control volume and  $\vec{n}_k$  is the outward pointing normal to the  $k^{\text{th}}$  edge scaled by its length.

The scalar numerical flux function of (2.9) is given by the generalisation of (2.6) to systems of equations, namely

$$\begin{aligned} (\underline{F}^*(\underline{U}_L, \underline{U}_R), \underline{G}^*(\underline{U}_L, \underline{U}_R)) \cdot \vec{n}_k &= \frac{1}{2} ((\underline{F}_L, \underline{G}_L) + (\underline{F}_R, \underline{G}_R)) \cdot \vec{n}_k \\ &\quad - \frac{1}{2} |(\tilde{\mathbf{A}}, \tilde{\mathbf{B}}) \cdot \vec{n}_k| (\underline{U}_R - \underline{U}_L), \end{aligned} \quad (3.4)$$

in which  $\mathbf{A} = \frac{\partial \underline{F}}{\partial \underline{U}}$  and  $\mathbf{B} = \frac{\partial \underline{G}}{\partial \underline{U}}$  are the flux Jacobians. The construction of  $\tilde{\mathbf{A}}$  and  $\tilde{\mathbf{B}}$ , the conservative approximations to the Jacobian matrices, and subsequently the numerical flux at the midpoint of the cell edge follows the technique suggested

by Roe [7]. The evolution of the discontinuous approximation to the solution is modelled by constructing a series of approximate Riemann problems at the edge midpoints with left and right states  $\underline{U}_L$  and  $\underline{U}_R$  respectively (the internal and external states relative to the control volume). Each Riemann problem is solved by decomposing the flux difference across the edge into its characteristic components, which results in a numerical flux function for edge  $k$  given by

$$\begin{aligned} (\underline{F}^*(\underline{U}_L, \underline{U}_R), \underline{G}^*(\underline{U}_L, \underline{U}_R)) \cdot \vec{n}_k &= \frac{1}{2} ((\underline{F}_L, \underline{G}_L) + (\underline{F}_R, \underline{G}_R)) \cdot \vec{n}_k \\ &\quad - \frac{1}{2} \sum_{j=1}^{N_w} \tilde{\alpha}^j |\tilde{\lambda}^j| \tilde{\underline{r}}^j. \end{aligned} \quad (3.5)$$

Here  $N_w$  is the number of components (or ‘waves’) in the decomposition, the symbol  $\tilde{\cdot}$  represents the Roe average value at the discontinuity (which is constructed so as to ensure that the linearised decomposition is conservative),  $\alpha^j$  is a wave ‘strength’ and  $\lambda^j$  and  $\underline{r}^j$ , respectively the eigenvalues and eigenvectors of the matrix  $(\mathbf{A}, \mathbf{B}) \cdot \vec{n}_k$ , represent the speed of the wave and the transformation of a perturbation of the characteristic variables into a perturbation of the conservative variables. Details of the exact values of these averages for the shallow water equations and the Euler equations are contained in Appendices A and B. The substitution of (3.5) into (3.3) together with the application of an appropriate time-stepping scheme gives the final algorithm.

One question remains: to which set of independent variables should the limiting procedures of Section 2.1 be applied? This is important because the choice affects the values of  $\underline{U}_L$ ,  $\underline{U}_R$ ,  $\underline{F}_L$  and  $\underline{F}_R$  in (3.4) and (3.5). The simplest approach is to apply the limiting to the conservative variables in a field by field manner (taking appropriate measures to ensure that negative pressures are not created by the reconstruction). It is also straightforward to limit the primitive variables via a simple transformation but, particularly since a characteristic decomposition is used in the calculation of the flux function (3.5), limiting the characteristic variables  $\underline{W}$  appears to be more appropriate.

Unfortunately, unlike the primitive variables, the characteristic variables  $\underline{W}$  are related to the conservative variables in terms of *differences*, that is

$$\Delta \underline{W} = \frac{\partial \widehat{W}}{\partial \underline{U}} \Delta \underline{U}, \quad (3.6)$$

so the limiting procedure must be recast in the same terms. Hence the limiting is carried out not on the values of the reconstructed solution at the midpoints of the cell edges, but on the differences between these values and the solution at the cell centroid.

The procedure defined in Section 2.1 is modified by redefining the vector  $\underline{P}_*$  in (2.12) to be

$$\underline{P}_* = \begin{pmatrix} x_* \\ y_* \\ \Delta w_* \end{pmatrix}, \quad (3.7)$$

where  $\Delta w_* = w_* - w_0$  and  $w$  is a component of the vector  $\underline{W}$  of characteristic variables. Each value of  $\Delta w$  is calculated using the formula (3.6) in which the approximate Jacobian  $\widehat{\frac{\partial \underline{W}}{\partial \underline{U}}}$  is evaluated at an appropriate average state, taken here to be the local approximation  $\frac{1}{2}(\underline{U}_0 + \underline{U}_*)$  at the relevant grid edge.

The limiting is applied to each component of  $\Delta \underline{W}$  in turn and the corresponding values of  $\Delta \underline{U}$  are retrieved by inverting (3.6). Therefore, on the occasions when no limiting is applied within a cell, the original values of  $\Delta \underline{U}$  are obtained.

It is unfortunate that advantage cannot be taken of the fact that the wave strengths in (3.5) are in fact differences in the characteristic variables, *i.e.*  $\alpha_j = \Delta W_j$ , but there appears to be no way of evaluating the Roe average states to give a conservative algorithm. Therefore, even though the limiting is applied to the characteristic variables, the conservative variables must be calculated before the approximate Riemann problems are constructed and solved.

### 3.1 Boundary Conditions

Simple characteristic boundary conditions are applied, in which the flux at a boundary edge is evaluated directly using information from within the boundary cell to supplement the imposed boundary values. The physical conditions applied at a given edge correspond to the positive eigenvalues of the matrix  $\mathbf{C} = \mathbf{A} \cos \theta + \mathbf{B} \sin \theta$ , where the conservative flux Jacobian matrices  $\mathbf{A}$  and  $\mathbf{B}$  are given in the appendices for the shallow water and Euler equations.

At a freestream boundary four possibilities pertain: (a) supercritical inflow, where all four eigenvalues are positive and the boundary flux is determined com-

pletely by the imposed solution values, (b) supercritical outflow, where no eigenvalue is positive and the flux is calculated from internal solution values, (c) subcritical inflow, where one eigenvalue is negative whose corresponding Riemann invariant is given its internal value and everything else is imposed, and finally (d) subcritical outflow, for which one eigenvalue is positive and the value of the associated Riemann invariant is imposed. At a solid wall the normal velocity component is set to zero while the rest of the information required to calculate the flux is taken from the interior of the domain.

This crude treatment of the boundaries has proved to be adequate for many of the cases considered, but in the future a ghost cell approach should be considered to improve the modelling.

## 3.2 Results

The first test case considered here is that of shallow water flow for a partial dam break problem [4]. The computational domain consists of a  $200m \times 200m$  basin bisected by a dam. When  $t = 0.0s$  a break in the dam appears between  $95m$  and  $170m$  from one end. Initially  $h = 10m$  on one side and  $h = 5m$  on the other, while the water has zero velocity everywhere. The 3688 cell grid on which the calculations were carried out is shown in Figure 3.1. Each of the boundaries is treated as a solid wall except those on the left and right.

Figures 3.2-3.3 show the surface elevation of the water at  $t = 7.2s$  for four schemes. The first order scheme is clearly the most diffusive of those shown but the other three give very similar results, particularly in the sharpness with which they capture the front of water moving downstream. Small differences appear in the minimum value of the depth of the water which occurs just downstream of the dam close to the edges of the gap where the flow is highly rotational. This is indicated by the slight differences in the scales on the vertical axes of the graphs. It is the MLG scheme which gives the lowest minimum while the LCD and GI schemes give roughly similar results.

Figure 3.4 shows the result obtained when the GI limiter is applied to the characteristic variables instead of the conservative variables. It is immediately obvious that the characteristic limiting is considerably more diffusive and there



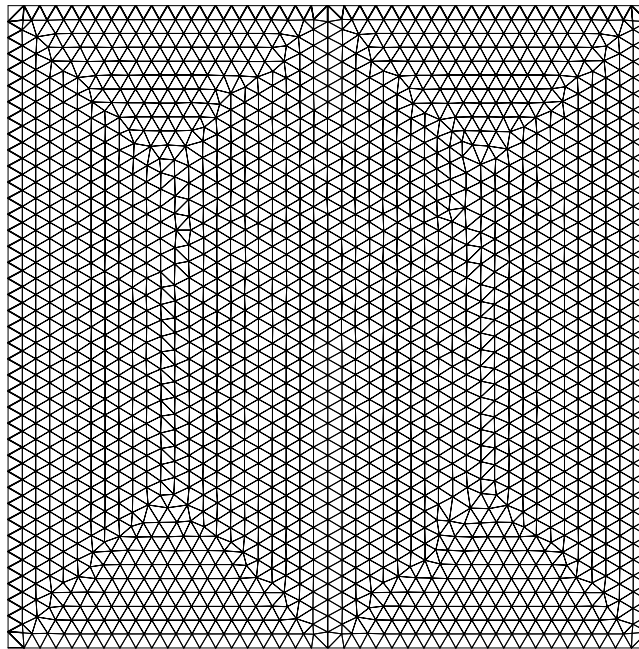


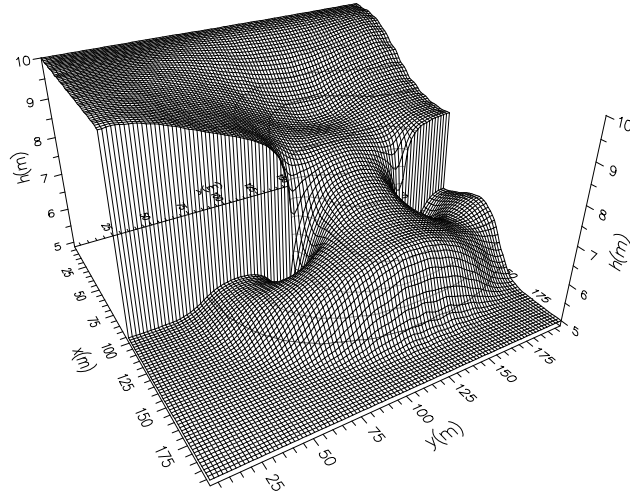
Figure 3.1: The grid for the partial dam break test case.

seems little reason to apply it.

A second dam break test case has been compared with experimental results [1]. This time a reservoir of water  $0.2m$  deep is released into an L-shaped channel with initial depth of  $0.01m$ . The velocity of the flow is zero everywhere when  $t = 0.0s$ . Bed friction effects are also modelled in this test case. A Manning roughness coefficient of  $0.0095$  was used and the appropriate source terms were treated in a simple node-by-node manner as described in [6]. The flow depth is sampled at the six points shown in Figure 3.5 at regular intervals until  $t = 41.0s$ . The figure also shows the geometry of the test case and the 2240 cell grid on which the solution has been approximated. Each boundary is treated as a solid wall except the one in the top right hand corner at which a supercritical outflow condition is applied. It should be noted that it was not possible to attain the exact initial conditions in the experiment, so the initial data for points P5 and P6 differs slightly from those of the numerical results.

The experimental results are pictures together with results obtained from the first order and LCD schemes. The boundary conditions proved to be inadequate for the more compressive limiters and the calculations blew up when the initial

### First order scheme



### LCD scheme

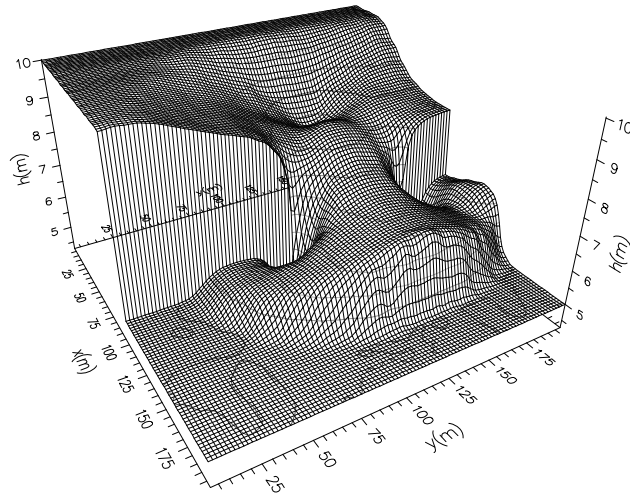
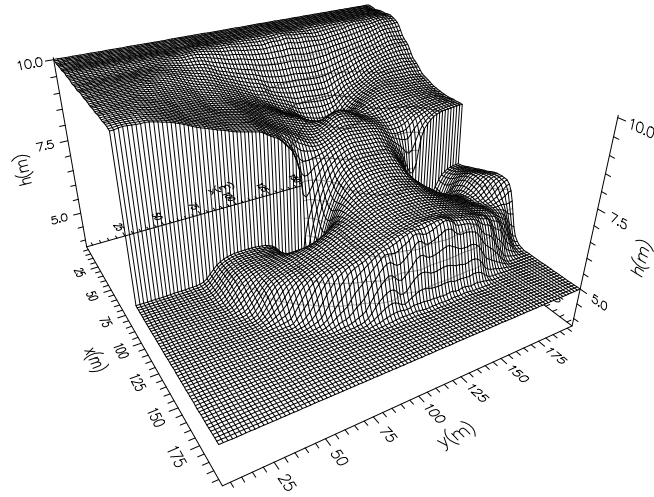


Figure 3.2: Solutions for the partial dam break test case.

### MLG scheme



### GI scheme

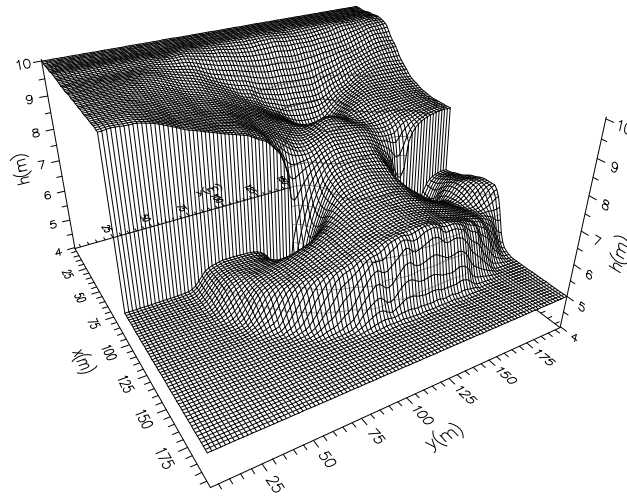


Figure 3.3: Solutions for the partial dam break test case.

## Characteristic GI scheme

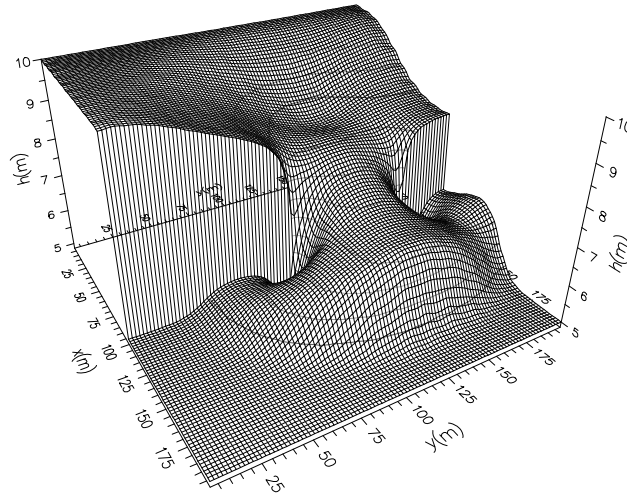


Figure 3.4: Solutions for the partial dam break test case when limiting is applied to the characteristic variables.

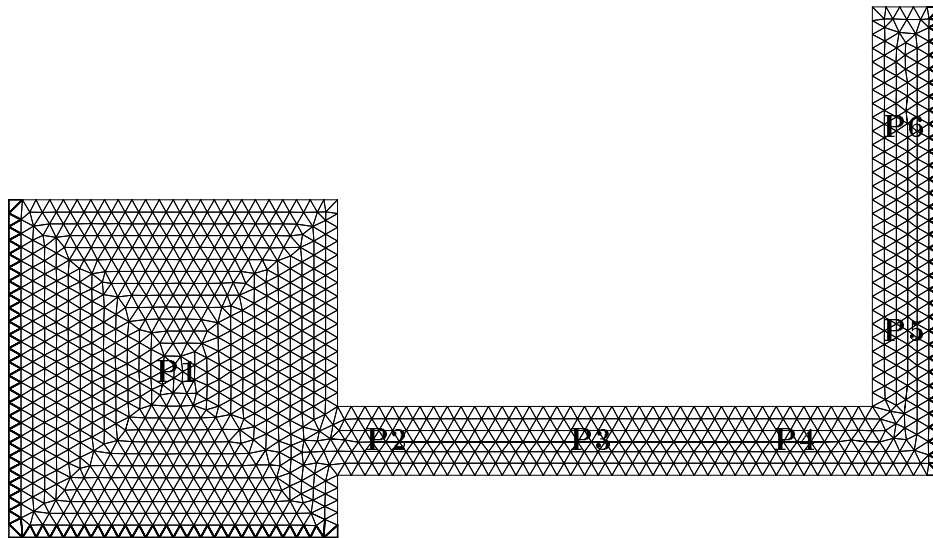


Figure 3.5: The grid and geometry for the dam break with L-shaped channel test case.

wave front reached the corner. The agreement with experimental data is good for both schemes at P1. At P2 the reflected wave front is captured more sharply and more accurately by the LCD scheme although neither agrees closely with experiment when the wave front first passes the point. The two numerical solutions are again in good agreement at P3 and P4, with the second order scheme capturing the fronts slightly more sharply. At P5 the LCD scheme gives a considerably better approximation to the experimental data while at P6 neither scheme gives particularly good agreement.

These schemes have also been applied to the Euler equations. One standard test case which has been used is that of Mach 3 flow over a forward facing step in a tunnel [11]. The tunnel is 3 units long and 1 unit wide and a step of height 0.2 is located 0.6 units into the tunnel. Initially the flow has density 1.4, pressure 1.0 and velocity 3.0 from left to right in Figure 3.7, which also shows the 2286 cell grid on which the problem was approximated. Both freestream boundaries have supercritical boundary conditions applied at them. The numerical solution obtained from the LCD scheme when  $t = 4.0$  is shown in Figure 3.8. It compares well with each of the other second order schemes illustrated in [11] on grids with similar numbers of cells, although the Mach stem on the step indicates that the modelling at the top corner of the step is more diffusive than desired. As in the previous test case the boundary conditions were not sufficiently robust to achieve solutions for the MLG or GI schemes.

## 4 Conclusions

In this report the construction of second order accurate monotonic cell centre finite volume schemes on triangular grids has been discussed. The methods are based on MUSCL-type schemes [10] in two dimensions in which a linear reconstruction of the solution is created within each cell from local data, the gradient of which is limited to impose monotonicity on the approximation. The methods have been tested on the scalar advection equation and then extended to nonlinear systems of equations via Roe's approximate Riemann solver. The extension of these schemes to general polygonal grids and three dimensions is also discussed

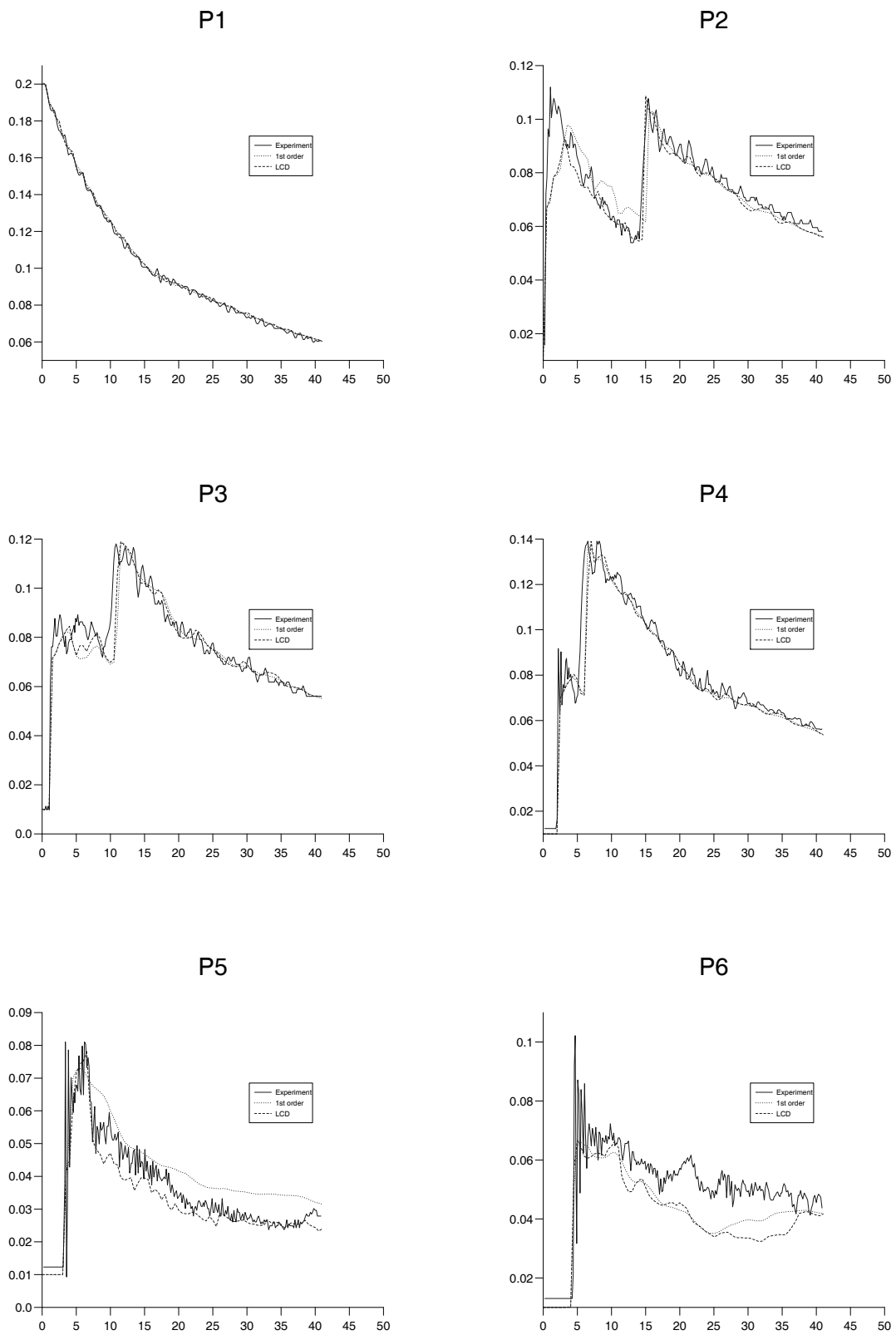


Figure 3.6: Comparison of experimental and numerical data for the dam break with L-shaped channel test case.

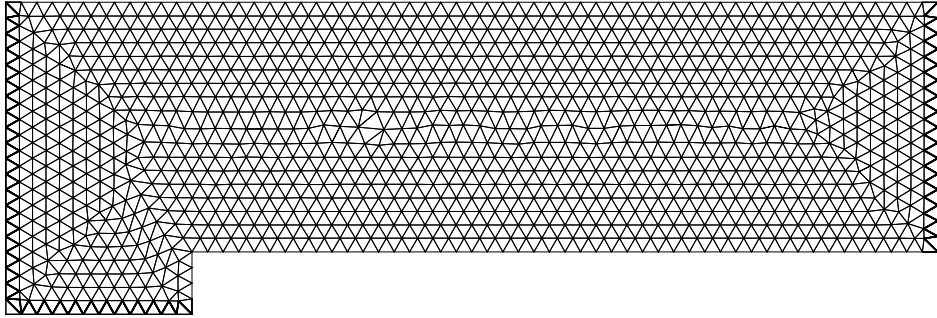


Figure 3.7: The grid for the forward facing step test case.

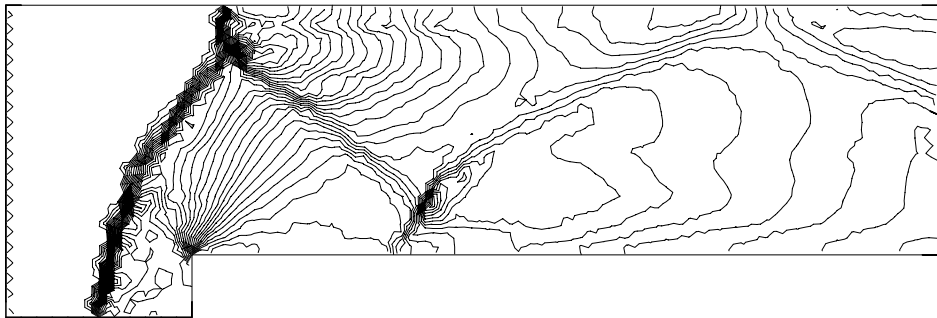


Figure 3.8: Density contours for the forward facing step test case.

briefly.

It has been shown that the family of monotonicity enforcing limiters can be completely defined by constraints applied at the midpoints of the edges of the cells. These constraints define a region within which every valid limiter lies. Furthermore, it follows that a limiter can be constructed which gives the maximum possible slope for the reconstruction. It has the property of removing the dependence of the reconstruction on the geometry of the surrounding grid cells but, although it is considerably cheaper than the most accurate of the previously constructed limiters and preserves peaks far better, the general quality of the solutions is worse. However, given the monotonicity region it should be possible to construct a more accurate limiter.

## Acknowledgements

The author would like to thank Prof. M. J. Baines for his contributions to this work and the EPSRC for providing the funding for the author.

## References

- [1] F.Alcrudo and P.Garcia-Navarro, ‘A high resolution Godunov-type scheme in finite volumes for the 2d shallow water equations’, *Int. J. for Num. Methods in Fluids*, **16**:489–505, 1993.
- [2] P.Batten, C.Lambert and D.M.Causon, ‘Positively conservative high-resolution convection schemes for unstructured elements’, *Int. J. for Num. Methods in Eng.*, **39**:1821–1838, 1996.
- [3] L.J.Durlofsky, B.Engquist and S.Osher, ‘Triangle based adaptive stencils for the solution of hyperbolic conservation laws’, *J. Comput. Phys.*, **98**:64–73, 1992.
- [4] R.J.Fennema and M.H.Chaudry, ‘Explicit methods for 2-D transient free-surface flows’, *J. Hydraul. Eng. ASCE*, **116**:1013–1034, 1990.



- [5] C.Hirsch, *Fundamentals of Numerical Discretization*, volume 1 of *Numerical Computation of Internal and External Flows*, Wiley, 1990.
- [6] M.E.Hubbard and M.J.Baines, ‘Conservative multidimensional upwinding for the steady two-dimensional shallow water equations’, To appear, *J. Comput. Phys.*, 1997.
- [7] P.L.Roe, ‘Approximate Riemann Solvers, parameter vectors, and difference schemes’, *J. Comput. Phys.*, **43(2)**:357–372, 1981.
- [8] P.K.Sweby, ‘High resolution schemes using flux limiters for hyperbolic conservation laws’, *SIAM J. Numer. Anal.*, **21**:995–1011, 1984.
- [9] B.van Leer, ‘Towards the ultimate conservative difference scheme V. A second order sequel to Godunov’s method’, *J. Comput. Phys.*, **32**:101–136, 1979.
- [10] B.van Leer, ‘On the relation between the upwind-differencing schemes of Godunov, Engquist-Osher and Roe’, *SIAM J. Scientific Statist. Comput.*, **5**, 1984.
- [11] P.R.Woodward and P.Collela, ‘The numerical simulation of two-dimensional fluid flow with strong shocks’, *J. Comput. Phys.*, **54**:115–173, 1984.

## A The Shallow Water Equations

The shallow water equations depend on the conservative variables and fluxes given by

$$\underline{U} = \begin{pmatrix} h \\ hu \\ hv \end{pmatrix}, \quad \underline{F} = \begin{pmatrix} hu \\ hu^2 + \frac{gh^2}{2} \\ huv \end{pmatrix}, \quad \underline{G} = \begin{pmatrix} hv \\ huv \\ hv^2 + \frac{gh^2}{2} \end{pmatrix}, \quad (\text{A.1})$$

where  $h$  is the depth of the flow,  $u$  and  $v$  are the  $x$ - and  $y$ -velocities and  $g$  is the acceleration due to gravity, and result in the following flux Jacobians:

$$\mathbf{A} = \frac{\partial \underline{F}}{\partial \underline{U}} = \begin{pmatrix} 0 & 1 & 0 \\ c^2 - u^2 & 2u & 0 \\ -uv & v & u \end{pmatrix}, \quad \mathbf{B} = \frac{\partial \underline{G}}{\partial \underline{U}} = \begin{pmatrix} 0 & 0 & 1 \\ -uv & v & u \\ c^2 - v^2 & 0 & 2v \end{pmatrix}, \quad (\text{A.2})$$

where  $c = \sqrt{gh}$  is the gravity wave speed.

In Roe's approximate Riemann solver the eigenvalues and eigenvectors of the matrix

$$(\mathbf{A}, \mathbf{B}) \cdot \vec{n} = \begin{pmatrix} 0 & n_x & n_y \\ (c^2 - u^2)n_x - uvn_y & 2un_x - vn_y & un_y \\ -uvn_x + (c^2 - v^2)n_y & vn_x & un_x + 2vn_y \end{pmatrix} \quad (\text{A.3})$$

are

$$\lambda_1 = \tilde{u}n_x + \tilde{v}n_y + \tilde{c}, \quad \lambda_2 = \tilde{u}n_x + \tilde{v}n_y, \quad \lambda_3 = \tilde{u}n_x + \tilde{v}n_y - \tilde{c}, \quad (\text{A.4})$$

and

$$\mathbf{r}_1 = \begin{pmatrix} 1 \\ \tilde{u} + \tilde{c}n_x \\ \tilde{v} + \tilde{c}n_y \end{pmatrix}, \quad \mathbf{r}_2 = \begin{pmatrix} 0 \\ -\tilde{c}n_y \\ \tilde{c}n_x \end{pmatrix}, \quad \mathbf{r}_3 = \begin{pmatrix} 1 \\ \tilde{u} - \tilde{c}n_x \\ \tilde{v} - \tilde{c}n_y \end{pmatrix}, \quad (\text{A.5})$$

respectively, and the corresponding wave strengths in (3.5) are given by

$$\begin{aligned} \tilde{\alpha}_1 &= \frac{\Delta h}{2} + \frac{1}{2\tilde{c}} (\Delta(hu)n_x + \Delta(hv)n_y - (\tilde{u}n_x + \tilde{v}n_y)\Delta h) \\ \tilde{\alpha}_2 &= \frac{1}{\tilde{c}} ((\Delta(hv) - \tilde{v}\Delta h)n_x - (\Delta(hu) - \tilde{u}\Delta h)n_y) \\ \tilde{\alpha}_3 &= \frac{\Delta h}{2} - \frac{1}{2\tilde{c}} (\Delta(hu)n_x + \Delta(hv)n_y - (\tilde{u}n_x + \tilde{v}n_y)\Delta h), \end{aligned} \quad (\text{A.6})$$

in which the Roe average states are

$$\tilde{u} = \frac{u_R\sqrt{h_R} + u_L\sqrt{h_L}}{\sqrt{h_R} + \sqrt{h_L}}, \quad \tilde{v} = \frac{v_R\sqrt{h_R} + v_L\sqrt{h_L}}{\sqrt{h_R} + \sqrt{h_L}}, \quad \tilde{c} = \sqrt{\frac{g(h_R + h_L)}{2}}, \quad (\text{A.7})$$

and the difference operator is given by

$$\Delta \cdot = (\cdot)_R - (\cdot)_L. \quad (\text{A.8})$$

In two dimensions the subscripts  $\cdot_L$  and  $\cdot_R$  represent the interior and exterior edge midpoint values relative to the cell under consideration.

## B The Euler Equations

The Euler equations depend on the conservative variables and fluxes given by

$$\underline{U} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ e \end{pmatrix}, \quad \underline{F} = \begin{pmatrix} \rho u \\ p + \rho u^2 \\ \rho uv \\ u(e + p) \end{pmatrix}, \quad \underline{G} = \begin{pmatrix} \rho v \\ \rho uv \\ p + \rho v^2 \\ v(e + p) \end{pmatrix}, \quad (\text{B.1})$$

where  $\rho$  is the density of the flow,  $u$  and  $v$  are the  $x$ - and  $y$ -velocities,  $p$  is pressure and  $e$  is the total energy, related to the other variables by an equation of state which, for a perfect gas, is

$$e = \frac{p}{\gamma - 1} + \frac{1}{2}\rho(u^2 + v^2), \quad (\text{B.2})$$

and result in the following flux Jacobians:

$$\mathbf{A} = \frac{\partial \underline{F}}{\partial \underline{U}} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ \frac{\gamma-1}{2}(u^2 + v^2) - u^2 & (3 - \gamma)u & (1 - \gamma)v & \gamma - 1 \\ -uv & v & u & 0 \\ \frac{\gamma-1}{2}u(u^2 + v^2) - uH & u^2(1 - \gamma) + H & (1 - \gamma)uv & \gamma u \end{pmatrix} \quad (\text{B.3})$$

and

$$\mathbf{B} = \frac{\partial \underline{G}}{\partial \underline{U}} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ -uv & v & u & 0 \\ \frac{\gamma-1}{2}(u^2 + v^2) - v^2 & (1 - \gamma)u & (3 - \gamma)v & \gamma - 1 \\ \frac{\gamma-1}{2}v(u^2 + v^2) - vH & (1 - \gamma)uv & v^2(1 - \gamma) + H & \gamma v \end{pmatrix}, \quad (\text{B.4})$$

where  $H = \frac{\gamma e}{\rho} - \frac{\gamma-1}{2}(u^2 + v^2)$  is the total enthalpy.

In Roe's approximate Riemann solver the eigenvalues and eigenvectors of the matrix

$$(\mathbf{A}, \mathbf{B}) \cdot \vec{n} = \mathbf{A}n_x + \mathbf{B}n_y \quad (\text{B.5})$$

are

$$\begin{aligned} \lambda_1 &= \tilde{u}n_x + \tilde{v}n_y + \tilde{c}, & \lambda_2 &= \tilde{u}n_x + \tilde{v}n_y, \\ \lambda_3 &= \tilde{u}n_x + \tilde{v}n_y, & \lambda_4 &= \tilde{u}n_x + \tilde{v}n_y - \tilde{c}, \end{aligned} \quad (\text{B.6})$$

and

$$\begin{aligned}
\mathcal{L}_1 &= \frac{\tilde{\rho}}{2\tilde{c}} \begin{pmatrix} 1 \\ \tilde{u} + \tilde{c}n_x \\ \tilde{v} + \tilde{c}n_y \\ \tilde{H} + \tilde{c}(\tilde{u}n_x + \tilde{v}n_y) \end{pmatrix}, \quad \mathcal{L}_2 = \tilde{\rho} \begin{pmatrix} 0 \\ -n_y \\ n_x \\ -\tilde{u}n_y + \tilde{v}n_x \end{pmatrix}, \\
\mathcal{L}_3 &= \begin{pmatrix} 1 \\ \tilde{u} \\ \tilde{v} \\ \frac{1}{2}(\tilde{u}^2 + \tilde{v}^2) \end{pmatrix}, \quad \mathcal{L}_4 = \frac{\tilde{\rho}}{2\tilde{c}} \begin{pmatrix} 1 \\ \tilde{u} - \tilde{c}n_x \\ \tilde{v} - \tilde{c}n_y \\ \tilde{H} - \tilde{c}(\tilde{u}n_x + \tilde{v}n_y) \end{pmatrix}, \quad (\text{B.7})
\end{aligned}$$

respectively, and the corresponding wave strengths in (3.5) are given by  $\alpha_j = \Delta W_j$  for  $j = 1, 2, 3, 4$ , where

$$\Delta \underline{W} = \mathbf{L} \Delta \underline{U} \quad (\text{B.8})$$

and

$$\mathbf{L} = \begin{pmatrix} -\frac{\tilde{u}n_x + \tilde{v}n_y}{\tilde{\rho}} + \frac{(\gamma-1)(\tilde{u}^2 + \tilde{v}^2)}{2\tilde{\rho}\tilde{c}} & \frac{n_x}{\tilde{\rho}} - \frac{(\gamma-1)\tilde{u}}{\tilde{\rho}\tilde{c}} & \frac{n_y}{\tilde{\rho}} - \frac{(\gamma-1)\tilde{v}}{\tilde{\rho}\tilde{c}} & \frac{(\gamma-1)}{\tilde{\rho}\tilde{c}} \\ \frac{\tilde{u}n_y - \tilde{v}n_x}{\tilde{\rho}} & -\frac{n_y}{\tilde{\rho}} & \frac{n_x}{\tilde{\rho}} & 0 \\ 1 - \frac{(\gamma-1)(\tilde{u}^2 + \tilde{v}^2)}{2\tilde{c}^2} & \frac{(\gamma-1)\tilde{u}}{\tilde{c}^2} & \frac{(\gamma-1)\tilde{v}}{\tilde{c}^2} & -\frac{(\gamma-1)}{\tilde{c}^2} \\ \frac{\tilde{u}n_x + \tilde{v}n_y}{\tilde{\rho}} + \frac{(\gamma-1)(\tilde{u}^2 + \tilde{v}^2)}{2\tilde{\rho}\tilde{c}} & -\frac{n_x}{\tilde{\rho}} - \frac{(\gamma-1)\tilde{u}}{\tilde{\rho}\tilde{c}} & -\frac{n_y}{\tilde{\rho}} - \frac{(\gamma-1)\tilde{v}}{\tilde{\rho}\tilde{c}} & \frac{(\gamma-1)}{\tilde{\rho}\tilde{c}} \end{pmatrix}. \quad (\text{B.9})$$

The Roe averages  $\tilde{u}$ ,  $\tilde{v}$ ,  $\tilde{c}$  and  $\tilde{H}$  are evaluated consistently from the average values of the parameter vector variables given by

$$\tilde{\underline{Z}} = \frac{1}{2}(\underline{Z}_L + \underline{Z}_R), \quad (\text{B.10})$$

where

$$\underline{Z} = \sqrt{\rho} \begin{pmatrix} 1 \\ u \\ v \\ H \end{pmatrix}. \quad (\text{B.11})$$